MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

MODULE TESTS

TRIMESTER 1, 2020/2021

ECE1016 – COMPUTER & PROGRAM DESIGN

(All Sections / Groups)

INSTRUCTIONS TO STUDENT

1. This question paper consists of 16 printed pages including cover page with 4 questions only.

2. Answer ALL the questions. All questions carry equal marks and the distribution of marks for each question is given.

3. Please write all your answers in the answer booklet provided.

Module Test 1

(a) A program written in C is converted into machine language using:

A. A code editor
B. A compiler
C. A terminal
D. An Integrated Development Environment (IDE)

[1 mark]

(b) Reorder the steps needed to implement a program.

i. Compile
ii. Repeat until program functions as expected
iii. Write source code
iv. Run program

A. i, ii, iii, and iv
B. ii, i, iv, and iii
C. iii, i, iv, and ii
D. iii, i, ii, and iv

[2 marks]

(c) Choose all the **valid** variable names from the list below:

i. `1Value`
ii. `ECE1016`
iii. `class size`
iv. `temp-celsius`
v. `_cake`
vi. `box_width`

A. i, ii, and iii
B. i, iii, and iv
C. ii, v, and vi
D. iii, iv, and vi
E. All of the above

[2 marks]

(d) What is the output of the following code snippet?

```
int main(){
    printf("\\nan");
    printf("\\bob");

    return 0;
}
```

A. `\`
`anob`
B. `\\nan\\bob`

C. `nanbob`
D. None of the above

[2 marks]

(e) What is the code used to produce the following output?

```
Sum is 57.000000
```

A. `printf("Sum is %f\n", '5' + 4.0);`
B. `printf("Sum is %f\n", 57 + 4.5);`
C. `printf("Sum is %s\n", "57.5555555");`
D. `printf("Sum is %c\n", 54 + 3);`

[1 mark]

(f) Which one of the following declaration is **incorrect**?

A. `str string;`
B. `char str;`
C. `int str;`
D. `char *str;`

[1 mark]

(g) Which one of the following initialisation is **incorrect**?

A. `char name[] = "John Smith";`
B. `char *name = "John Smith";`
C. `char name[100] = "John Smith";`
D. `char *name[100] = "John Smith";`

[1 mark]

(h) Write a C statement to accomplish each of the following:

(i) Declare a pointer `numPtr` that points to an integer variable.

[1 mark]

(ii) Assign the pointer to the address of a variable named `num`.

[1 mark]

(iii) Print out the value pointed by `numPtr`.

[1 mark]

(i) Analyse the program shown below. Select the appropriate code snippets for each of the missing lines of code. The output of the program is also shown below.

```
1   Line 1
2
3   int main(){
4
5       int num1 = 9;
6       int num2 = 5;
7          Line 7
8       int *Ptr2;
```

```
 9
10      Ptr1 = &num1;
11      num2 = *Ptr1 * num2;
12
13      Ptr2 = &num2;
14          Line 14
15
16      printf("*Ptr1 = %d\n", *Ptr1);
17          Line 17
18
19      return 0;
20  }
```

```
*Ptr1 = 45
*Ptr2 = 45
```

Line 1:

A. `int *Ptr1`
B. `int *xyz`
C. `printf("*Ptr2 = %d\n", *Ptr2);`
D. `Ptr1 = Ptr2;`
E. `printf("*Ptr2 = %d\n", Ptr2);`
F. `Ptr1 = *Ptr2;`
G. `#include <stdio.h>`
H. `#include <studio.h>`

Line 7:

A. `int *Ptr1`
B. `int *xyz`
C. `printf("*Ptr2 = %d\n", *Ptr2);`
D. `Ptr1 = Ptr2;`
E. `printf("*Ptr2 = %d\n", Ptr2);`
F. `Ptr1 = *Ptr2;`
G. `#include <stdio.h>`
H. `#include <studio.h>`

Line 14:

A. `int *Ptr1`
B. `int *xyz`
C. `printf("*Ptr2 = %d\n", *Ptr2);`
D. `Ptr1 = Ptr2;`
E. `printf("*Ptr2 = %d\n", Ptr2);`
F. `Ptr1 = *Ptr2;`
G. `#include <stdio.h>`
H. `#include <studio.h>`

Line 17:

A. `int *Ptr1`

B. `int *xyz`
C. `printf("*Ptr2 = %d\n", *Ptr2);`
D. `Ptr1 = Ptr2;`
E. `printf("*Ptr2 = %d\n", Ptr2);`
F. `Ptr1 = *Ptr2;`
G. `#include <stdio.h>`
H. `#include <studio.h>`

[4 marks]

(j) Provide the output of the program shown below at the following lines:

```
1  #include <stdio.h>
2
3  int main(){
4      char a[5] = "honda";
5
6      printf("%c\n",*a);
7      printf("%c\n",*(a+3));
8
9      return 0;
10 }
```

(i) Line 6

[1 mark]

(ii) Line 7

[1 mark]

(k) Assuming `str1` contains the string "`COVID`", `str2` contains the string "`19`", and `str3` contains the string "`virus`". Provide the output for the following C statements:

(i) `printf("%s-%s %s", str1, str2, str3);`

[1 mark]

(ii)
```
if (strcmp(str3, "vaccine")!=0){
    printf("Continue to social distance");
}
else
    printf("Cure found");
```

[2 marks]

(iii) `printf("%c", *(str1+2));`

[1 mark]

(iv) `printf("%d\n", strlen(strcat(str1, str2)));`

[2 marks]

Module Test 2

(a) What is the keyword to transfer control from a function back to the calling function?

A. switch
B. return
C. goto
D. back

[1 mark]

(b) Consider the following function `multiply`. What is the output when the function is called as `printf("%d ", multiply(2,3))` in the `main` function?

```
int multiply(int x, int y){
    x = x*y;
    return y;
}
```

A. 6
B. 2
C. 3
D. None of the above

[2 marks]

(c) What does the following function `func` do?

```
float func(int num){
    int i;
    float f=1;

    for(i=1; i<=num; i++)
        f = f * i;
    return f;
}
```

A. The function calculates the square root of an integer
B. The function calculates the factorial value of an integer
C. The function calculates the value of 1 raised to the power `num`
D. None of the above

[2 marks]

(d) Every function in the C program must be used more than once.

A. True
B. False

[2 marks]

(e) Each function is executed in the order in which it is defined in the program source file.

A. True
B. False

[2 marks]

(f) Rewrite the following mathematical expressions using the predefined C functions in `math.h` header file.

(i) $\sqrt{(x-y)^2}$

[2 marks]

(ii) $\log_{10}(a^b)$

[2 marks]

(g) Figure MT1(a) shows an incomplete C program while the output of the program is shown in Figure MT2(b). Read the program carefully, and answer the following questions.

```
#include <stdio.h>

int circle(float a, float b);
void display(float a, float b, int *c);

int main() {

   float x, y;
   int unitcircle;

   printf("Please enter x-coordinate: ");
   scanf("%f", &x);
   printf("Please enter y-coordinate: ");
   scanf("%f", &y);

   unitcircle = circle(x,y);
   display(x,y,&unitcircle);

   return 0;
}

<MISSING FUNCTION DEFINITIONS GO HERE>
```

Figure MT2(a): Program to calculate the unit circle

```
Please enter x-coordinate: 3
Please enter y-coordinate: 4
Point (3.00, 4.00) is outside of the unit circle
```

```
Please enter x-coordinate: 0.3
Please enter y-coordinate: 0.4
```

```
Point (0.30, 0.40) is inside of the unit circle
```

Figure MT2(b): Sample output for 2 different inputs. Underlined denote user's input.

Based on the incomplete program shown in Figure MT1(a):

(i) What are the function prototypes in the source code?

[2 marks]

(ii) Identify 'call-by-value' and 'call-by-reference' functions.

[2 marks]

(iii) Complete the function definition of `circle()` which takes two input parameters `x` and `y`. This function determines whether the point (`x,y`) lies inside the unit circle (the circle with centre at the origin and radius 1). If the point is inside the unit circle ($x^2 + y^2 < 1$), the function will return `1`, otherwise it will return `0`.

[4 marks]

(iv) Complete the function definition of `display()` which takes three input parameters, `x`, `y`, and a pointer `c`. This function displays the result on the screen depending on the value of `unitcircle`. If it is 1, the function will display *Point (x,y) is inside of the unit circle*. Otherwise, *Point (x,y) is outside of the unit circle* will be displayed on the screen.

[Note: Refer to Figure MT2(b) for the output format]

[4 marks]

Module Test 3

(a) Which of the following file types can be opened using `fopen()` function in C?

i. .txt
ii. .bmp
iii. .dat
iv. .bin
v. .c

A. i and iv
B. i, iv, v
C. ii, iii, and iv
D. i, ii, iii and iv
E. All of the above

[2 marks]

(b) Why are File operations (e.g. `fprintf`, `fscanf`, `fopen`, etc.) recommended when handling data?

A. Volatile memory will automatically delete data, hence improving security.
B. It is a benchmark of a professional programmer.
C. Data stored in non-volatile storage are safer and more practical.
D. It is more practical to store data in memory (RAM).

[2 marks]

(c) What is a structure in C language?

A. A structure is a collection of elements that can be of different data type.
B. A structure is a collection of elements that can be of same data type.
C. Elements of a structure are called members.
D. All of the above.

[2 marks]

(d) The '−>' operator can be used to access structures elements using a pointer.

A. True
B. False

[1 mark]

(e)

```
#include <stdio.h>

//Answer for part (i)
//Definition of struct Fruit

int main(){

    int i, count = 0;

    //Answer for part (ii)
    //Declare and initialise array called inventory

    //Answer for part (iii)
    //Display the discounted price of fruits

    //Answer for part (iv)
    //Display fruit(s) with low inventory

    printf("%d items are running low-stock\n", count);

    return 0;
}
```

Figure MT3(a): Program Listing

Fruit	Price	Quantity
Grape	23.50	35
Apple	12.90	70
Banana	6.70	55
Orange	8.80	45

Table MT3(a): Inventory of available fruits

You are required to write a program to store the record of the available fruits in a fruit store. Based on the Program Listing in Figure MT3(a), answer the following questions:

(i) Define a suitable structure called `Fruit` that can be used to store the fruit, price, and quantity of available fruits.

[3 marks]

(ii) Declare a new array called `inventory` of the type `struct Fruit` and initialise it with the data in Table 1.

[3 marks]

(iii) The store owner would like to give a 30% discount on all fruits. Write the C code to print out the discounted prices of each fruit on the screen, as shown in Figure MT3(b) below.

```
New price for Grape is RM 16.45.
New price for Apple is RM 9.03.
```

```
New price for Banana is RM 4.69.
New price for Orange is RM 6.16.
```

Figure MT3(b): Output

[3 marks]

(iv) The store owner would like to know which fruits have less than 50 units left in stock. Write the C code to display the fruit(s).

```
Running low: Grape
Running low: Orange
```

Figure MT3(c): Output

[3 marks]

(f) Identify **FOUR (4)** lines with errors in the code listed in Figure MT3(d). For each line with error, write down the line number followed by the corrected version in your answer. For example:

Line 1: #include <stdio.h>

```
1  #include <stdio.c> //Ignore this. There are 4 others
2
3  int main()
4  {
5        FILE fp;
6        fp = fopen("Grades.txt", r+);
7
8        fseek(*fp,0,SEEK_END);
9        fputs("George was here", *fp);
10
11       fclose(fp);
12
13       return 0;
14 }
```

Figure MT3(d)

[6 marks]

Module Test 4

(a) Which of the following code snippets will produce the following result?

```
0 0 0 0 0
1 1 1 1 1
2 2 2 2 2
3 3 3 3 3
4 4 4 4 4
```

A.

```
int i, j;

for (i=0;i<5;i++)
{
    for(j=5;j>0;j--)
    {
        printf("%d ", j);
    }
    printf("\n");
}
```

B.

```
int i, j;

for (i=0;i<5;i++)
{
    for(j=0;j<=i;j++)
    {
        printf("%d ", j);
    }
    printf("\n");
}
```

C.

```
int i=0, j;

while(i<5)
{
    j=0;
    while(j<5)
    {
        printf("%d ", i);
        j++;
    }

    printf("\n");
    i++;
}
```

D.

```
int i=5, j;

while(i>0)
{
    j=0;
    while(j<i)
    {
        printf("%d ", j);
        j++;
    }

    printf("\n");
    i--;
}
```

[2 marks]

(b) Which one of the following statements best describe this program?

```
#include <stdio.h>

int main()
{
    int a = 2;

    switch(a)
    {
        case a > 0:
            printf("Positive\n");
            break;
        case a < 0:
            printf("Negative\n");
            break;
        case a == 0:
            printf("Zero\n");
            break;
    }

    return 0;
}
```

A. The program runs as expected and displays “Zero”.
B. The program runs as expected and displays “Positive”.
C. The program does not run because of compilation errors.
D. The program runs but does not function as expected because of logic errors.

[2 marks]

(c) What is the output of the following program?

```
#include <stdio.h>

int main()
{
    int x = 1;

    if (x > 0)
        printf("Option A\n");
    else if (x > 0)
        printf("Option B\n");
    else if (x == 1)
        printf("Option C\n");
    else
        printf("Option D\n");

    return 0;
}
```

A. Option A
B. Option B
C. Option C
D. Option D

[2 marks]

(d) Determine whether the outcome of the following relational expressions are TRUE or FALSE.

(i) `!5==0`

[1 mark]

(ii) `(5<2) || !(4>=6)`

[1 mark]

(iii) `(5!=5) || (10>=4) && !(-13<0)`

[1 mark]

(e) A 2-dimensional array is declared and initialised as below:

```
int TwoD[m][n] = {{10, 20},{30,40,50},{60,70},{80}};
```

(i) What is the value for `m`?

[1 mark]

(ii) What is the value for `n`?

[1 mark]

(iii) What is the syntax to access the value 30?

[1 mark]

(iv) What is the syntax to assign the value of 99 to `TwoD[1][1]`?

[1 mark]

(v) What is the value of `TwoD[1][2]`?

[1 mark]

(vi) What is the value of `TwoD[2][2]`?

[1 mark]

(f) Write a complete C program that records and analyses the number of COVID-19 cases over a week (Day 1 to Day 7). The program should (i) prompt the user to key in the number of cases for each day **using a loop**, (ii) find the maximum and minimum number of cases and their corresponding days, (iii) determine whether the number of cases is increasing or decreasing on Day 7 compared to Day 1, and (iv) the corresponding percentage increment or decrement.

To achieve the **FOUR (4)** objectives, you can use the following suggestions (not compulsory).

(i) Using a loop, the program prompts the user to key in the number of cases for 7 days and **stores them in an array**. Assume that the user will always key in zero or more cases, therefore the program does not need to check the inputs.

(ii) Assign one of the 7 number of cases as the minimum and then compare it with the others. If there is a smaller number, replace the minimum with the smaller number. Record the corresponding day (i.e. Day 1, 2, …, 7) as well. Repeat the same process for maximum cases.

(iii) Compare the number of cases on Day 7 with Day 1. If Day 7 has more cases, then display "Increasing", else if there are less cases, display "Decreasing", or else "No changes".

(iv) Calculate the percentage increment/decrement using the formula below:

$$\frac{Day\ 7 - Day\ 1}{Day\ 1} \times 100$$

A sample of the output is shown in Figure MT4(a) below.

```
Cases for Day 1: 77
Cases for Day 2: 66
Cases for Day 3: 55
Cases for Day 4: 44
Cases for Day 5: 33
Cases for Day 6: 22
Cases for Day 7: 11

Max cases:   77 on Day 1
Min cases:   11 on Day 7
Changes: Decreasing
Percentage change: -85.71%
```

Figure MT4(a): **Bold and underline** denote user's input.

[10 marks]